COBRTL

```
CCCCCCCC    000000    BBBBBBBB    SSSSSSSS   WW      WW   IIIIII    TTTTTTTTTT   CCCCCCCC   HH      HH
CCCCCCCC    000000    BBBBBBBB    SSSSSSSS   WW      WW   IIIIII    TTTTTTTTTT   CCCCCCCC   HH      HH
CC          00    00  BB    BB    SS         WW      WW     II          TT       CC         HH      HH
CC          00    00  BB    BB    SS         WW      WW     II          TT       CC         HH      HH
CC          00    00  BB    BB    SS         WW      WW     II          TT       CC         HH      HH
CC          00    00  BB    BB    SS         WW      WW     II          TT       CC         HH      HH
CC          00    00  BBBBBBBB    SSSSSS     WW      WW     II          TT       CC         HHHHHHHHHH
CC          00    00  BBBBBBBB    SSSSSS     WW      WW     II          TT       CC         HH      HH
CC          00    00  BB    BB          SS   WW  WW  WW     II          TT       CC         HH      HH
CC          00    00  BB    BB          SS   WW  WW  WW     II          TT       CC         HH      HH
CC          00    00  BB    BB          SS   WWWW  WWWW     II          TT       CC         HH      HH  ....
CC          00    00  BB    BB          SS   WWWW  WWWW     II          TT       CC         HH      HH  ....
CCCCCCCC    000000    BBBBBBBB    SSSSSSSS   WW      WW   IIIIII        TT       CCCCCCCC   HH      HH  ....
CCCCCCCC    000000    BBBBBBBB    SSSSSSSS   WW      WW   IIIIII        TT       CCCCCCCC   HH      HH  ....

LL          IIIIII    SSSSSSSS
LL          IIIIII    SSSSSSSS
LL            II      SS
LL            II      SS
LL            II      SS
LL            II      SS
LL            II      SSSSSS
LL            II      SSSSSS
LL            II            SS
LL            II            SS
LL            II            SS
LL            II            SS
LLLLLLLLLL  IIIIII    SSSSSSSS
LLLLLLLLLL  IIIIII    SSSSSSSS
```

```
   1   0001  0 MODULE COBSSWITCH(
   2   0002  0                  IDENT = '1-006'        ! file: COBSWITCH.B32  EDIT:LGB1006
   3   0003  0                  ) =
   4   0004  1 BEGIN
   5   0005  1
   6   0006  1 !
   7   0007  1 !****************************************************************
   8   0008  1 !*                                                              *
   9   0009  1 !*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                    *
  10   0010  1 !*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.     *
  11   0011  1 !*   ALL RIGHTS RESERVED.                                       *
  12   0012  1 !*                                                              *
  13   0013  1 !*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
  14   0014  1 !*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
  15   0015  1 !*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
  16   0016  1 !*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
  17   0017  1 !*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
  18   0018  1 !*   TRANSFERRED.                                               *
  19   0019  1 !*                                                              *
  20   0020  1 !*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
  21   0021  1 !*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
  22   0022  1 !*   CORPORATION.                                               *
  23   0023  1 !*                                                              *
  24   0024  1 !*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
  25   0025  1 !*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.    *
  26   0026  1 !*                                                              *
  27   0027  1 !*                                                              *
  28   0028  1 !****************************************************************
  29   0029  1 !
  30   0030  1 !++
  31   0031  1 ! FACILITY:     COBOL SUPPORT
  32   0032  1 !
  33   0033  1 ! ABSTRACT
  34   0034  1 !
  35   0035  1 !     Reads switches
  36   0036  1 !
  37   0037  1 !
  38   0038  1 ! ENVIRONMENT:  Vax-11 User Mode
  39   0039  1 !
  40   0040  1 ! AUTHOR: MLJ , CREATION DATE: 16-JAN-79
  41   0041  1 !
  42   0042  1 ! MODIFIED BY:
  43   0043  1 !
  44   0044  1 ! 1-001 - Original. MLJ  16-JAN-79
  45   0045  1 ! 1-002 - Added boilerplate and comments. RKR 23-AUG-1979
  46   0046  1 ! 1-003 - Change symbolic name of LIBRARY file. RKR 1-OCT-79
  47   0047  1 ! 1-004 - Cosemtic changes.  RKR 18-OCT-79
  48   0048  1 ! 1-005 - Fix typo from 4.  RKR 20-OCT-79
  49   0049  1 ! 1-006 - Use logical LNMSC_NAMLENGTH instead of hard-coded value of 64.
  50   0050  1 !                                          LGB 31-OCT-1983
  51   0051  1 !--
  52   0052  1
  53   0053  1 !<BLF/PAGE>
```

```
55      0054    1  !
56      0055    1  !  SWITCHES
57      0056    1  !
58      0057    1
59      0058    1  SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
60      0059    1
61      0060    1  !
62      0061    1  !  LINKAGES
63      0062    1  !
64      0063    1  LINKAGE
65      0064    1      L = JSB : GLOBAL(PTR=2, CNT=3, DIG=4);
66      0065    1  !
67      0066    1  !  TABLE OF CONTENTS:
68      0067    1  !
69      0068    1  FORWARD ROUTINE
70      0069    1
71      0070    1        GET_DIGIT : L ;
72      0071    1
73      0072    1  !  INCLUDE FILES
74      0073    1
75      0074    1  REQUIRE 'RTLIN:RTLPSECT' ;               ! Macros to declare psects
76      0169    1  LIBRARY 'RTLSTARLE';
77      0170    1
78      0171    1  !
79      0172    1  !  MACROS
80      0173    1  !      NONE
81      0174    1  !
82      0175    1  !
83      0176    1  !  EQUATED SYMBOLS
84      0177    1  !
85      0178    1  !      NONE
86      0179    1  !
87      0180    1  !  PSECT DECLARATIONS:
88      0181    1  !
89      0182    1
90      0183    1  DECLARE_PSECTS (COB) ;                   ! Declare psects for COB$ facility
91      0184    1
```

```
 93    0185  1  GLOBAL ROUTINE COB$SWITCH(NUMBER)=
 94    0186  1
 95    0187  1  !++     FUNCTIONAL DESCRIPTION
 96    0188  1
 97    0189  1
 98    0190  1  !  See C79RTLREQ for specification
 99    0191  1  !
100    0192  1  !
101    0193  1  !  FORMAL PARAMETERS:
102    0194  1  !
103    0195  1  !      NUMBER
104    0196  1  !
105    0197  1  !  IMPLICIT INPUTS:
106    0198  1  !
107    0199  1  !      Logical name table
108    0200  1  !  IMPLICIT OUTPUTS:
109    0201  1  !
110    0202  1  !      NONE
111    0203  1  !
112    0204  1  !
113    0205  1  !  ROUTINE VALUE:
114    0206  1  !      1       Success
115    0207  1  !      0       Failure
116    0208  1  !
117    0209  1  !  COMPLETION CODES:
118    0210  1  !
119    0211  1  !      NONE
120    0212  1  !
121    0213  1  !  SIDE EFFECTS:
122    0214  1  !
123    0215  1  !      NONE
124    0216  1  !
125    0217  1  !--
126    0218  1  !
127    0219  2      BEGIN
128    0220  2      LOCAL
129    0221  2          BUFFER:             VECTOR[LNM$C_NAMLENGTH,BYTE], ! Returned equivalence string
130    0222  2          BUFDESC:            VECTOR[2],                   ! Descriptor for BUFFER
131    0223  2          NAMDESC:            VECTOR[2],                   ! Descriptor for logical name
132    0224  2          LENGTH:             WORD;                        ! Length of equivalence string
133    0225  2      GLOBAL REGISTER
134    0226  2          PTR=2, CNT=3, DIG=4;
135    0227  2
136    0228  2
137    0229  2      BUFDESC[0] = LNM$C_NAMLENGTH;                        ! Set up descriptors
138    0230  2      BUFDESC[1] = BUFFER;                                 !     ...
139    0231  2      NAMDESC[0] = 12;                                     !     ...
140    0232  2      NAMDESC[1] = UPLIT('COB$SWITCHES');                  !     ...
141    0233  2
142    0234  2
143  P 0235  2      IF NOT $TRNLOG(                                      ! Translate COB$SWITCHES
144    0236  2          LOGNAM=NAMDESC, RSLLEN=LENGTH, RSLBUF=BUFDESC)
145    0237  2      THEN
146    0238  2          RETURN 0;
147    0239  2
148    0240  2
149    0241  2      PTR = BUFFER;                                       ! Set up global registers
```

```
150    0242  2              CNT = .LENGTH;                          !    ...
151    0243
152    0244
153    0245                 WHILE 1 DO
154    0246                     BEGIN LOCAL VAL;
155    0247
156    0248                     ! Skip leading non-digits
157    0249
158    0250                     WHILE (IF .CNT GTR 0 THEN NOT GET_DIGIT() ELSE RETURN 0) DO 0;
159    0251
160    0252                     ! Remember first digit
161    0253
162    0254                     VAL = .DIG;
163    0255
164    0256                     ! Scan over digits to next non-digit, computing value in VAL
165    0257
166    0258                     WHILE (IF .CNT GTR 0 THEN GET_DIGIT() ELSE 0) DO VAL = .VAL * 10 + .DIG;
167    0259
168    0260                     ! Check VAL against desired switch
169    0261
170    0262                     IF .VAL EQL .NUMBER THEN RETURN 1;
171    0263  2              END;
172    0264
173    0265
174    0266  2          RETURN 0;
175    0267  1      END;
```

```
                                           .TITLE  COB$SWITCH
                                           .IDENT  \1-006\

                                           .PSECT  _COB$CODE,NOWRT,  SHR,  PIC,2

  53  45  48  43  54  49  57  53  24  42  4F  43  00000 P.AAA:  .ASCII  \COB$SWITCHES\       :

                                           .EXTRN  SYS$TRNLOG

                          OFFC 00000        .ENTRY  COB$SWITCH, Save R2,R3,R4,R5,R6,R7,R8,R9,-   : 0185
                                                    R10,R11
                    5E  FEEC  CE  9E 00002   MOVAB   -276(SP), SP
           0C  AE   FF    8F  9A 00007       MOVZBL  #255, BUFDESC                             : 0229
           10  AE   14    AE  9E 0000C       MOVAB   BUFFER, BUFDESC+4                         : 0230
           04  AE         0C  D0 00011       MOVL    #12, NAMDESC                              : 0231
           08  AE   DC    AF  9E 00015       MOVAB   P.AAA, NAMDESC+4                          : 0232
                    7E    7C 0001A           CLRQ    -(SP)                                     : 0236
                    7E    D4 0001C           CLRL    -(SP)
              18  AE    9F 0001E             PUSHAB  BUFDESC
              10  AE    9F 00021             PUSHAB  LENGTH
              18  AE    9F 00024             PUSHAB  NAMDESC
   00000000G  00        06  FB 00027         CALLS   #6, SYS$TRNLOG
              32        50  E9 0002E         BLBC    R0, 4$
              52   14    AE  9E 00031        MOVAB   BUFFER, PTR                               : 0241
              53        6E  3C 00035         MOVZWL  LENGTH, CNT                               : 0242
                        53  D5 00038 1$:     TSTL    CNT                                       : 0250
                        27  15 0003A         BLEQ    4$
                   0000V 30 0003C           BSBW    GET_DIGIT
           F6             50  E9 0003F       BLBC    R0, 1$
```

```
                              55      54 D0 00042        MOVL    DIG, VAL          0254
                                      53 D5 00045 2$:    TSTL    CNT               0258
                                      10 15 00047        BLEQ    3$
                                   0000V 30 00049        BSBW    GET_DIGIT
                              0A      50 E9 0004C        BLBC    R0, 3$
                    50        55      0A C5 0004F        MULL3   #10, VAL, R0
                    55        50      54 C1 00053        ADDL3   DIG, R0, VAL
                                      EC 11 00057        BRB     2$
                 04 AC                55 D1 00059 3$:    CMPL    VAL, NUMBER       0262
                                      D9 12 0005D        BNEQ    1$
                    50                01 D0 0005F        MOVL    #1, R0
                                      04 00062           RET
                                      50 D4 00063 4$:    CLRL    R0                0267
                                      04 00065           RET
```

; Routine Size: 102 bytes,   Routine Base: _COBSCODE + 000C

;   176   0268 1
;   177   0269 1 !<BLF/PAGE>

```
 179    0270  1  ROUTINE GET_DIGIT: L=
 180    0271  1  !
 181    0272  1  !++        FUNCTIONAL DESCRIPTION
 182    0273  1  !
 183    0274  1  !
 184    0275  1  !  Get next character from string, returning:
 185    0276  1  !              PTR advanced to next character
 186    0277  1  !              CNT decremented
 187    0278  1  !              DIG containing the character less %C'0'
 188    0279  1  !              Value true if and only if the character was a digit
 189    0280  1  !
 190    0281  1  !
 191    0282  1  !  FORMAL PARAMETERS:
 192    0283  1  !
 193    0284  1  !
 194    0285  1  !  IMPLICIT INPUTS:
 195    0286  1  !
 196    0287  1  !      NONE
 197    0288  1  !
 198    0289  1  !  IMPLICIT OUTPUTS:
 199    0290  1  !
 200    0291  1  !      NONE
 201    0292  1  !
 202    0293  1  !  ROUTINE VALUE:
 203    0294  1  !  COMPLETION CODES:
 204    0295  1  !
 205    0296  1  !      NONE
 206    0297  1  !
 207    0298  1  !  SIDE EFFECTS:
 208    0299  1  !
 209    0300  1  !      NONE
 210    0301  1  !
 211    0302  1  !--
 212    0303  1  !
 213    0304  2      BEGIN
 214    0305  2      EXTERNAL REGISTER
 215    0306  2          PTR, CNT, DIG;
 216    0307  2      LOCAL
 217    0308  2          VAL;
 218    0309  2
 219    0310  2      VAL = 0;                              ! Assume false value
 220    0311  2      DIG = .(.PTR)<0,8>;                   ! Fetch character
 221    0312  2      PTR = .PTR + 1;                       ! Advance pointer
 222    0313  2      CNT = .CNT - 1;                       ! Decrement count
 223    0314  2      DIG = .DIG - %C'0';                   ! Bias character
 224    0315  2      IF .DIG LEQU 9 THEN VAL = .VAL + 1;   ! Set true value if need be
 225    0316  2      .VAL                                  ! Return the value
 226    0317  1      END;
```

```
              50  D4 00000 GET_DIGIT:
                                       CLRL    VAL                  ; 0310
         54   82  9A 00002             MOVZBL  (PTR)+, DIG          ; 0311
              53  D7 00005             DECL    CNT                  ; 0313
```

```
                        54          30 C2 00007        SUBL2   #48, DIG              : 0314
                        09          54 D1 0000A        CMPL    DIG, #9              : 0315
                                    02 1A 0000D        BGTRU   1$
                                    50 D6 0000F        INCL    VAL
                                    05 00011 1$:       RSB                           : 0317
```

; Routine Size: 18 bytes,   Routine Base: _COB$CODE + 0072

; 227   0318 0 END ELUDOM

                              PSECT SUMMARY
;
;
;   Name   Bytes   Attributes
;
; _COB$CODE   132 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

;   Library Statistics
;
;   -------- Symbols --------   Pages   Processing
;   File   Total   Loaded Percent   Mapped   Time
;
; _255DUA28:[SYSLIB]STARLET.L32;1   9776   4   0   581   00:00.7

;   COMMAND QUALIFIERS
;
;   BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:COBSWITCH/OBJ=OBJ$:COBSWITCH MSRC$:COBSWITCH/UPDATE=(ENH$:COBSWITCH
;   )

; Size: 120 code + 12 data bytes
; Run Time: 00:03.5
; Elapsed Time: 00:15.8
; Lines/CPU Min: 5451
; Lexemes/CPU-Min: 20811
; Memory Used: 62 pages
; Compilation Complete

COBRMSBLO
LIS

COBSPANC
LIS

COBSWITCH
LIS

COBRESTVA
LIS

CONVSHR
MAP

CONVCOMIO
LIS

COBSETSWI
LIS

CONVDEF
MDL

CONVLINK
REQ

COBSPANC2
LIS

CONV

CONVERT
MAP

CONVCALL
LIS

COBVECTOR
LIS

CONVERT
REQ

CONV

RECLAIM
MAP

ADDKEY
LIS